RECEIVED
CENTRAL FAX CENTER
MAR 2 2 2007

Remarks

This Amendment is responsive to the **January 26, 2007** Final Office Action. Reexamination and reconsideration of the remaining claims (1-11, and 14-22) is respectfully requested. A telephonic interview is also requested and the Examiner is invited to contact the undersigned attorney. Agenda items for the telephonic interview appear within and include discussing how the claims distinguish over the reference.

Summary of The Office Action

Claims 1-13 were rejected under 35 U.S.C. §112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. Amendments have been made to overcome these rejections.

Claims 1-4, 6-21, and 23-24 were rejected under 35 U.S.C. §102(e) as being anticipated by Bhatia et al. (US Patent No. 6,535,798 B1)(Bhatia). Amendments have been made to the remaining claims to clarify how these claims distinguish over Bhatia. Additionally, arguments are presented that describe how the claimed subject matter differs from Bhatia. For example, Bhatia is an ACPI system that produces actual ACPI state changes while the claims describe a system that is not an ACPI system and that produces simulated ACPI states.

Claims 5 and 22 were rejected under 35 U.S.C. §103(a) as being unpatentable over Bhatia. Amendments have been made to the remaining claims to distinguish over Bhatia. Additionally, arguments are presented that describe how the claimed subject matter differs from Bhatia. For example, Bhatia is an ACPI system that produces actual ACPI state changes while the claims describe a system that is not an ACPI system and that produces simulated ACPI states.

Claims 1-24 were rejected under 35 U.S.C. §101 as being directed to non-statutory subject matter. The definitions of logic and computer-readable medium have been amended

to overcome this rejection. The Office Action asserts that software is not patentable. This is simply incorrect. The PTO can not overrule case law like Beauregard. In re Beauregard, 53 F. 2d 1583, 35 USPQ 2d 1382 (Fed. Cir. 1995), see also, in re Lowry, 32 F. 3d 1579, 32 USPQ 2d 1031 (Fed. Cir. 1994) Claim 22 is in proper Beauregard format.

Objections to the Drawings

The Office Action indicates that Figures 1-8 should be designated by a legend such as -- Prior Art -- because they purportedly illustrate only that which is old. Applicant respectfully disagrees since the figures illustrate systems and methods that help explain and understand the present invention. Therefore, they are not prior art and should not be labeled as such. The objection to the drawings should be withdrawn.

Docket No. 200314976-1

35 U.S.C. §101

Claims 1-24 were rejected under 35 U.S.C. §101 as purportedly being directed to non-statutory subject matter. The Office Action asserts that "software per se" is not patentable. The claims are not directed to "software per se", but rather to an apparatus to change processor states, a method to change processor states, and to a computer readable medium.

The Examiner is encouraged to examine the claims as an apparatus, a method, and as a computer-readable medium and not to apply a blanket 101 rejection to anything that could possibly be "software per se". The recent spate of "software per se" and "functional descriptive language" rejections produced by the Patent Office are reminiscent of the "technological arts test" rejections once produced by the Patent Office. Therefore, the Examiner is encourage to read Ex parte Lundgren, Appeal No. 2003-2088 (BPAI 2005), to discover how such creations are likely to be treated on appeal. Ex parte Lundgren makes clear that the PTO must follow the law and not create the law. The law is that apparatus, methods, and computer readable mediums are patentable, even when they include executing software, and particularly when claimed in proper Beauregard format.

The claims withstand the incorrect application of the law by the Office Action. The claimed invention passes the practical utility test because it provides a useful, concrete, and tangible result by changing the providing of a clock signal to a processor to produce a change in a processor without actually making an ACPI processor state change. Analyzing the processor before and after the method was executed would reveal a change in performance. This is no less useful, concrete, and tangible than a process that opens a mold on a rubber curing machine. The result is useful because it facilitates actions like thermal control. The result is concrete (e.g., repeatable) because selecting a bit pattern from the data structure and writing the selected bit pattern to the ACPI throttling register will produce the throttling result in a repeatable fashion. The result is tangible (e.g., discernible, measurable) and can be easily detected using, for example, a protocol analyzer, a logic probe, or an oscilloscope. Thus the claims survive even in light of the erroneous interpretation of 35 U.S.C. §101.

The Claims Patentably Distinguish Over the References of Record

35 U.S.C. §102

For a 35 U.S.C. §102 reference to anticipate a claim, the reference must teach every element of the claim. Section 2133 of the MPEP recites:

> A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference. Verdegaal Bros. v. Union Oil Co. of California, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987).

Here, the reference does not teach producing a simulated processor state. The reference simply teaches making an actual ACPI processor state change in a processor having ACPI elements, a variable voltage supply, and a variable frequency clock.

Bhatia

Bhatia describes a true reactive ACPI system that works with processors that have elements including internal machine specific registers, variable voltage supplies, variable frequency clocks, and so on. Thus, Bhatia describes how to create a true ACPI processor state in a processor by manipulating one or more of the voltage and frequency of a processor. Interestingly, this appears to be patentable subject matter because Bhatia is a patent.

The application describes simulating a processor state (e.g., voltage, frequency) in a processor that does not have internal machine specific registers, variable voltage supplies, variable frequency clocks, and so on. The processor state is simulated by selectively throttling the processor by selecting and writing throttling bit patterns to an ACPI throttling register.

Bhatia describes storing "the location and structure of the control register" in an ACPI object C12, l51-53 along with "the number of performance states available, the core clock frequencies and supply level voltages to be used in performance states, the expected power consumption in each performance state", and so on. What is missing from the ACPI object are the **claimed set of throttling bit patterns that are used in throttling to simulate the results of changing frequencies and supply level voltages** and the logic to select between the members of the set of bit patterns to simulate a state. These bit patterns are missing because Bhatia establishes true states, it does not simulate them.

The values stored in Bhatia are the "core clock frequencies" and "supply level voltages" that can be used to control processors having variable voltage supplies and variable frequency clocks. Since those elements are not available in the processors upon which the claims operate, those types of values are not stored. Instead, values for controlling throttling are stored.

Bhatia clearly recites that ACPI objects may be created and used. C12, 126-58. However, no mention is made of an object or data structure that stores **both** the address of an ACPI throttling register and throttling bit patterns to write to the ACPI throttling register. The addresses do not need to be stored because Bhatia is an ACPI system in which the addresses are already known. Throttling bit patterns are not stored because power dissipation is controlled in decrements of ΔP, a binary action for which no bit pattern storage is required. C9, 143-45.

35 U.S.C. §103

To establish a prima facie case of 35 U.S.C. §103 obviousness, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. MPEP 2143.01 Second, there must be a reasonable expectation of success. MPEP 2143.02 Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. MPEP 2143.03 Additionally, the teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art, not in applicant's disclosure. In re Vaeck, 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991). This requirement is intended to prevent unacceptable "hindsight reconstruction" where Applicant's invention is recreated from references using the Application as a blueprint.

Here, the third criteria described in MPEP 2143.03 is not satisfied since the reference does not teach or suggest all the claim limitations. The reference does not teach simulating a processor state without actually creating the processor state. Additionally, the reference does not teach simulating the processor state in a processor that has neither a variable voltage

supply nor a variable frequency clock. Thus, none of the claims are obvious for at least this reason.

Claims will now be discussed individually.

Independent Claim 1

Claim 1 has been amended so that the features upon which the Applicant bases the arguments are found explicitly in the claim. Claim 1 has been amended to remove any optional language.

Claim 1 is directed to an apparatus that includes a memory and a logic. The memory stores throttling values and the logic processes these values to produce a novel, concrete, tangible, and useful result. The result is producing a change in how a processor operates to simulate ACPI state changes without causing an actual ACPI state change. Rather than sending a ΔP signal to move through available ACPI states, throttling bit patterns are stored, read, and written. An actual ACPI state change may not be possible because the processor with which the claimed apparatus interacts may not have a variable voltage supply and/or variable frequency supply. Adding the claimed apparatus to a non-ACPI capable system may facilitate providing ACPI-like services in the system.

Claim 1 recites a memory that stores both throttling bit patterns to write to an ACPI throttling register and the address of the ACPI throttling register. Claim 1 also recites selecting between the stored bit patterns and writing the selected bit pattern to the ACPI throttling register to simulate a processor state. The reference does not disclose selectively writing selected throttling bit patterns to an ACPI throttling register to simulate a processor state without causing an actual ACPI state change. The reference discloses sending ΔP signals to move through available ACPI states.

Differences between a true processor state and a simulated processor state are discussed in [0014] of the application:

> Implementing a true processor performance state may include changing an internal clock frequency for a processor, changing a voltage at which a processor will operate and so on. Simulating a processor performance state may include using an ACPI accessible throttling register to throttle a processor. Throttling a processor

Docket No. 200314976-1

> may include, for example, controlling the percentage of time during which a processor clock operates and/or controlling the percentage of time during which a processor is supplied with a clock signal. In a true processor performance state, a clock frequency may change. In a simulated processor performance state, the clock frequency for the processor may remain substantially the same but the throttling register may partially and/or completely disable the clock and/or block the clock signal from being supplied to a processor thus controlling the number of clock edges seen by a processor. This facilitates simulating the frequency change associated with a true processor performance state. Thus, the frequency change is achieved without using a machine specific register internal to a processor as is typical in true processor performance state systems.

Differences between creating a true processor state and a simulated processor state are also discussed in [0035]:

> However, the processor **210** may not include internal machine specific registers, variable voltage supplies, variable frequency clocks and so on. Thus, rather than producing an actual processor performance state in the processor **210**, the system **200** may produce a simulated processor performance state by configuring the throttling register **230** to cause the processor **210** to be throttled. The processor **210** may be throttled, for example, when a signal is asserted on the STOPCLK# line.

The reference describes changing frequency and/or voltage to create an actual processor state. The claim describes throttling to simulate a processor state. Thus, the claim distinguishes over the reference and is in condition for allowance.

Since claim 1 recites features not taught or suggested by the reference, claim 1 patentably distinguishes over the reference. Accordingly, dependent claims 2-11 also patentably distinguish over the reference and are in condition for allowance.

Claims 2-11

These claims depend from claim 1, which has been shown to be not anticipated. Thus, these claims are similarly not anticipated.

Docket No. 200314976-1

Claim 5

Claim 5 was rejected as being unpatentable over Bhatia. This claim describes establishing the ACPI table in a BIOS. Bhatia describes how "thermal management in the computer system 10 may be accomplished by other modules including software, firmware, and/or hardware modules. ... the thermal management routine may be implemented in another software layer (e.g., an OS module, device driver, BIOS routine)." C7, 155-61. While this describes how the management **routine** may be stored in a BIOS, it does not describe the **table** being part of the BIOS or being established in the BIOS by the apparatus. No table may be required because no set of values is needed to toggle a line to do ΔP processing.

The Office Action repeats several times that "the thermal management routine **includes** the data structure, therefore disclosing implementing an ACPI table in a BIOS." This assertion cites column 3, lines 15-19 for support. The assertion is incorrect and the citation does not support the assertion.

The assertion is incorrect because Bhatia does not disclose establishing an ACPI table in a BIOS. In Bhatia, if an ACPI table exists in a BIOS, it was burnt in there by some other process. Since it would already be in the BIOS, there would be no need to establish it in the BIOS in order to simulate the processor states. Also since a ΔP approach is taken, no table is required to store throttling values that are not used in Bhatia.

Word by word analysis of the citation, and indeed of the entire reference reveals that the purported support for the incorrect assertion is missing. During the telephonic interview requested by Applicant, one agenda item is to identify where in this passage Bhatia indicates that **the routine includes the data structure.** Another agenda item will be to have the Examiner explain how the executable instructions in Bhatia are patentable while the executable instructions in the Application are not patentable.

Text	Supports "routine includes data structure"?
In one embodiment	No
the thermal management	No
may be performed	No

Docket No. 200314976-1

in an ACPI environment	No
It is to be understood, however	No
that other power or thermal management schemes	No
may also be used	No
to achieve thermal management	No
while maintaining a relatively high level	No
of system performance	No

Claim 7

This claim depends from claim 1, which has been shown to be not anticipated and thus this claim is similarly not anticipated. Additionally, this claim has been amended to make clear that the processor for which the processor state is simulated does not have a variable voltage supply. Bhatia can only work in a processor having a variable voltage supply and/or variable frequency source. For this additional reason this claim is not anticipated and is in condition for allowance.

Claim 9

This claim depends from claim 8, which has been shown to be not anticipated and thus this claim is similarly not anticipated. Additionally, this claim recites that the set of bit patterns facilitates simulating processor performance states by throttling the processor for different percentages of time. While Bhatia describes establishing different processor performance states, they are actual performance states, not simulated performance states.

The Office Action asserts that C8, 155-61 teaches this limitation. This passage reads: "At one extreme, TC1 can be set to zero and change in performance ΔP may be completely based on the difference between T_n and T_l. According to one embodiment, ΔP may be set at 12.5% increments". While the reference describes 12.5% increments, it does not describe throttling bit patterns that are used to establish the eight simulated processor states having the specifically called out values in the application. In Bhatia, to the extent that the specified states would be achieved, they would be achieved through means that did not include

selecting a bit pattern and writing the bit pattern. More likely a signal to increase or decrease by ΔP would be provided. This signal would not specify the desired simulated state, just whether to increase or decrease performance. For this additional reason this claim is not anticipated and is in condition for allowance.

Independent Claim 14

Claim 14 is directed to a method for simulating a processor state. Bhatia is directed towards systems and methods for creating true processor states. Claim 14 recites selecting a bit pattern to write to an ACPI throttling register. Bhatia recites storing true clock frequencies and voltages for processors having variable voltage sources and variable clock frequencies. Where Bhatia recites throttling, it describes changing performance by a ΔP amount achieved through a binary signal, not through writing one of a stored bit pattern. For at least these reasons claim 14 is not anticipated and is in condition for allowance.

Claims 15-21

These claims depend from claim 14, which has been shown to be not anticipated, and thus claims are similarly not anticipated.

Claim 15

This claim also describes establishing the data structure as an ACPI table in a BIOS operably connectable to the processor. Bhatia describes how "thermal management in the computer system 10 may be accomplished by other modules including software, firmware, and/or hardware modules. ... the thermal management routine may be implemented in another software layer (e.g., an OS module, device driver, BIOS routine)." C7, 155-61. While this describes how the **management routine** may be stored in a BIOS, it does not describe the **table** being part of the BIOS because, quite simply, no such table is used in Bhatia. Furthermore, it does not describe **establishing** the table in a BIOS. For this additional reason this claim is not anticipated and is in condition for allowance.

Docket No. 200314976-1

Claim 16

This claim further characterizes establishing the data structure as an ACPI table in a BIOS operably connectable to the processor. Additional limitations include writing a set of bit patterns to the ACPI table and writing the address of the ACPI throttling register to the ACPI table. The Office Action asserts that C12, lines 40-43 disclose this limitation. However, this passage recites: "This may be performed, for example, by writing a predefined value to a control register to indicate the new performance state." Writing a value to a **control register** does not teach writing to a **data structure** (e.g., ACPI table) both a set of throttling bit patterns and the ACPI throttling register address to the same data structure. For this additional reason this claim is not anticipated and is in condition for allowance.

Conclusion

For the reasons set forth above, **claims 1-11 and 14-22** patentably and unobviously distinguish over the references and are allowable. An early allowance of all claims is earnestly solicited. Applicant hereby requests a telephonic interview to provide an opportunity to discuss how the claims distinguish over the reference.

Respectfully submitted,

John T. Kalnay (Reg. No. 46,816)
(216) 348-5844
McDonald Hopkins, LLC
600 Superior Avenue, E.
Suite 2100
Cleveland, OH 44114